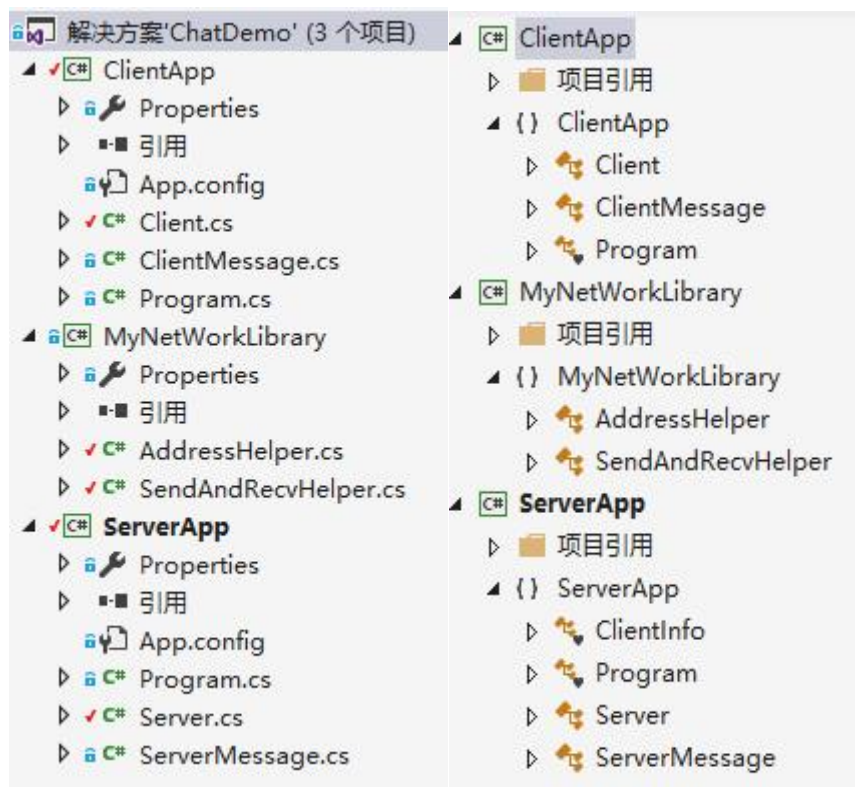


# ChatDome 设计文档

## 1 程序的运行环境、安装步骤

(1) 运行环境: .Net Framework 4.5.2

(2) 程序的组成部份:



ClientApp 为客户端, ServerApp 为服务器端, ClientMessage.cs 里是客户端的提示信息, ServerMessage.cs 里是服务器端的提示信息, 这些信息用来在客户在输入不规范的命令, 或不当的操作(直接关闭)时给出的提示信息。MyNetWorkLibrary 是一个辅助项目, 用来处理 socket 发送接受数据, 以及 IPEndPoint 的获取。

(3) 安装步骤:

1) 安装 .Net Framework 4.5.2。

2) 将程序 AChatServer 文件夹复制到计算机上

3) 在资源管理器中，双击 server.exe，运行服务器端，双击 ChatDemo.exe，运行程序客户端

## 2 程序开发平台

(1) 代码行数: 400

(2) 开发环境: Visual studio community 2015

## 3 程序功能说明:

```
Server 10.10.32.70 is listening port 53223...

Please enter the server ip: 10.10.32.70
Please enter the server port: 53223
Please enter your name to login in: lucy
Welcome lucy!
[Me] /create dinner
[Me] /list
dinner
kk has joined
[kk] hi everyone
[Me] hi kk!
[Me] nice
[Me]

Please enter the server ip: 10.10.32.70
Please enter the server port: 53223
Please enter your name to login in: kk
Welcome kk!
[Me] /join dinner
[Me] hi everyone
[lucy] hi kk!
[lucy] nice
[Me]
```

主要功能参考自 UCB 计算机网络课程 CS168 的 Project1:

[https://github.com/NetSys/cs168\\_student/blob/master/projects/proj1\\_chat/spec.md](https://github.com/NetSys/cs168_student/blob/master/projects/proj1_chat/spec.md)

如上图示例所示，首先启动 server，在启动若干个 client，输入 server 的 ip 和 port，输入名字。

所有的输入分为 2 中，一种普通输入，会被广播到所有同频道的 client，另一种命令以 '/' 开头，支持 3 中命令 /join, /list, /create，所有命令都有格式要求，错误的格式会出现响应的错误提示信息。用户加入或退出频道会有提示信息。

## 4 程序算法说明及面向对象实现技术方案

(1) 对用到的算法的说明，简要介绍其算法步骤、时间空间复杂度等

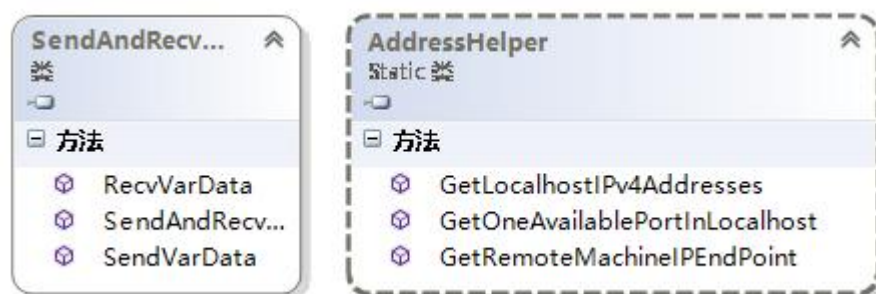
维护了一个 connectionSocket, name, channel 之间关系的数据结构, connectionSocket 与 name 一一对应, 一个 channel 可以有多个 connectionSocket, 同时如果某个 connectionSocket 没有 channel, 则默认 channel 为 "", 在此 channel 中信息不广播。这些对应关系都是使用 Dictionary 来实现的。

还有一个存放 name 的 List。

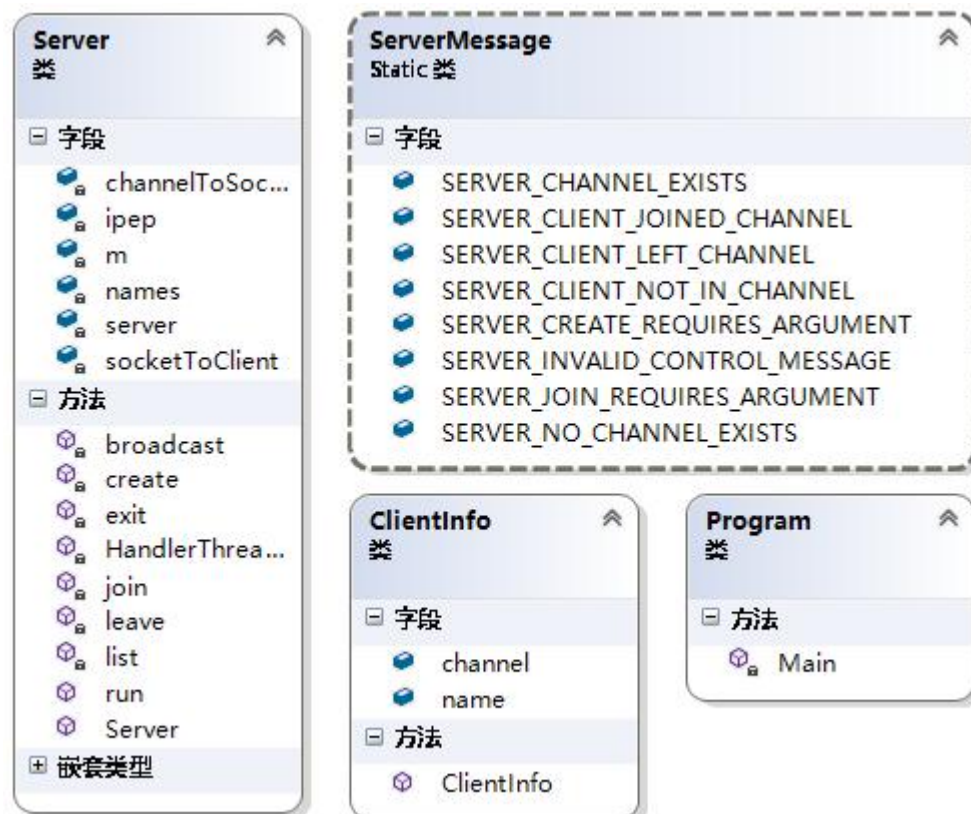
所有的对此数据结构的改动都只是对 Dictionary 的 add 和 remove, 故时间复杂度在  $O(1)$ 。

## (2) 数据结构和算法的面向对象实现

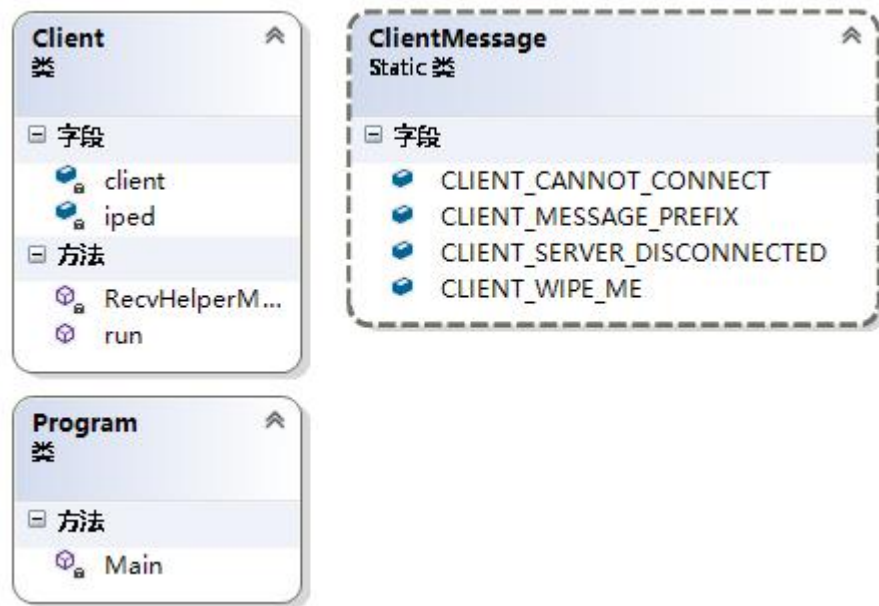
MyNetWorkLibrary 项目：



Server 项目：



Client 项目：



## 5 技术亮点、关键点及其解决方案

### ● 本程序的亮点：

我写的程序支持多个客户端同时连接，无阻塞！

程序支持发送和接受任意长数据

### ● 本程序的技术关键点

设计了 **MyNetWorkLibrary** 项目帮助发送和接收数据，规定发送和接收的数据开头 4 个字节为数据大小。

我采用了多线程技术实现多个客户端无阻塞的通信。

我使用 Mutex 来同步数据，使得一次只能有一个线程修改其中的数据结构。

### ● 遇到的技术难点及对应的解决方案：

处理客户端或者服务器端异常关闭的情况

**问题描述：**当用户直接关闭用户端时，辅助类 **MyNetWorkLibrary** 会抛出 `socketExpection` 异常，而关于此情况错误的信息提示在 `server` 中。

**最终的解决方案：**在 **MyNetWorkLibrary** 中捕获此异常，并重新抛出，在 `server` 中处理此异常。

## 6 简要开发过程

- |          |  |
|----------|--|
| 5 月 24 号 | 查找资料确定要使用的算法与数据结构                      |
| 5 月 24 号 | 完成系统设计                                 |
| 5 月 25 号 | 进行 MyNetWorkLibrary 的发送和接受数据功能开发       |
| 5 月 25 号 | MyNetWorkLibrary 的发送和接受数据功能开发完成，进入测试阶段 |
| 5 月 26 号 | 程序主体功能开发完成                             |
| 6 月 2 号  | 程序完成处理各种错误信息的功能，增加程序的鲁棒性               |
| 6 月 3 号  | 程序开发工作完毕，编写及整理文档                       |